Amendments to the Claims

1. (currently amended) A system for detecting and restricting denial of service attacks, comprising:

a transmit ~~algorithm~~ module to receive outgoing packets from a software application and to discard the outgoing packets that are determined to be from a zombie application prior to being transmitted over a network;

a receive ~~algorithm~~ module to receive incoming packets from a network interface and to discard the incoming packets that are determined to be from a zombie application; and

a monitor ~~code~~ module in communications with the transmit ~~algorithm~~ module and the receive ~~algorithm~~ module to track transmit ~~the pattern of~~ packet ~~transmission~~ patterns from and receive packet patterns to ~~reception to and from~~ the software application and to determine ~~that~~ whether the software application is ~~a~~ the zombie application based upon the ~~pattern of~~ transmit and receive packet ~~transmission and reception~~ patterns.

2. (currently amended) The system recited in claim 1, wherein ~~said~~ the monitor ~~code~~ module determines that the software application is the zombie application by identifying that the software application is transmitting a large number of packets without receiving any packets and placing the software application on a zombie list or a watch list.

3. (currently amended) The system recited in claim 2, wherein ~~said~~ the monitor ~~code~~ module alerts the user and the transmit ~~algorithm~~ module and the receive ~~algorithm~~ module that ~~a~~ the software application is ~~a~~ the zombie application when the software application has previously been placed on the zombie list or the watch list and the software application is still transmitting a large number of packets without receiving any packets.

4. (currently amended) The system recited in claim 1, wherein ~~said~~ the monitor ~~code~~ module determines that the software application is ~~the~~ a possible zombie application by identifying that the software application is not receiving any packets and placing the software application on a watch list.

5. (currently amended) The system recited in claim 4, wherein ~~said~~ the monitor ~~code~~ module alerts the user and the transmit ~~algorithm~~ module and the receive ~~algorithm~~ module that ~~a~~ the software application is ~~a~~ the zombie application when the software application has previously been placed on the watch list and the software application is now transmitting a large number of packets.

6. (currently amended) The system recited in claim 1, wherein ~~said~~ the monitor ~~code~~ module determines that the software application is a possible zombie application by identifying that the software application is rarely receiving any packets and placing the software application on a watch list.

7. (currently amended) The system recited in claim 6, wherein ~~said~~ the monitor ~~code~~ module alerts the user and the transmit ~~algorithm~~ module and the receive ~~algorithm~~ module that ~~a~~ the software application is ~~a~~ the zombie application when the software application has previously been placed on the watch list and the software application is now transmitting a large number of packets.

8. (currently amended) The system recited in claim 1, wherein ~~said~~ the monitor ~~code~~ module determines that the software application is a possible zombie application by identifying that the software application, after having received some packets~~,~~ has stopped sending ~~receiving any~~ packets or receiving more packets and placing the software application on a watch list.

9. (currently amended) The system recited in claim 8, wherein ~~said~~ the monitor ~~code~~ module alerts the user and the transmit ~~algorithm~~ module and the receive ~~algorithm~~ module that ~~a~~ the software application is ~~a~~ the zombie application when the software application has previously been placed on the watch list and the software application is now transmitting a large number of packets.

10. (currently amended) The system recited in claim 3, wherein ~~said~~ the monitor ~~code~~ module will retain ~~an~~ the software application on the watch list when a zombie rating for the software application exceeds a predetermined value, ~~said~~ the zombie rating ~~is~~ being based on the factors of whether the software application is an application or a

process and whether the software application is user initiated or initiated at system startup.

11. (currently amended) The system recited in claim 5, wherein ~~said~~ the monitor ~~code~~ module will retain ~~an~~ the software application on the watch list when a zombie rating for the software application exceeds a predetermined value, ~~said~~ the zombie rating ~~is base~~ being based on the factors of whether the software application is an application or a process and whether the software application is user initiated or initiated at system startup.~~.~~

12. (currently amended) The system recited in claim 7, wherein ~~said~~ the monitor ~~code~~ module will retain ~~an~~ the software application on the watch list when a zombie rating for the software application exceeds a predetermined value, ~~said~~ the zombie rating ~~is base~~ being based on the factors of whether the software application is an application or a process and whether the software application is user initiated or initiated at system startup.

13. (currently amended) The system recited in claim 9, wherein ~~said~~ the monitor ~~code~~ module will retain ~~an~~ the software application on the watch list when a zombie rating for the software application exceeds a predetermined value, ~~said~~ the zombie rating ~~is base~~ being based on the factors of whether the software application is an application or a process and whether the software application is user initiated or initiated at system startup.

14. (currently amended) A method of detecting and restricting denial of service attacks, comprising:

monitoring incoming and outgoing packets to and from a software application;

placing ~~said~~ the software application on a zombie list or a watch list when a pattern of the incoming or outgoing packets to or from the software applications matches that of the characteristics of a zombie application; and

blocking reception and transmission of packets to and from the software application when the software application has been placed on the watch list or the zombie list in a previous cycle and the software application further exhibits the characteristics of a zombie application.

15. (original) The method recited in claim 14, wherein the characteristics of a zombie application are transmitting a large number of packets while receiving no incoming packets.

16. (original) The method recited in claim 14, wherein the characteristics of a zombie application are receiving no incoming packets and having a zombie rating exceeding a predetermined value.

17. (currently amended) The method recited in claim 16, wherein ~~said~~ the zombie rating is based on the factors of whether the software application is an application or a

process and whether the software application is user initiated or initiated at system startup.

18. (original) The method recited in claim 14, wherein the characteristics of a zombie application are rarely receiving incoming packets and having a zombie rating exceeding a predetermined value.

19. (currently amended) The method recited in claim 18, wherein ~~said~~ the zombie rating is based on the factors of whether the software application is an application or a process and whether the software application is user initiated or initiated at system startup.

20. (original) The method recited in claim 14, wherein the characteristics of a zombie application are receiving incoming packets at first and then not receiving or sending any packets and having a zombie rating exceeding a predetermined value.

21. (currently amended) The method recited in claim 20, wherein ~~said~~ the zombie rating is based on the factors of whether the software application is an application or a process and whether the software application is user initiated or initiated at system startup.

22. (currently amended) A computer program, comprising:

monitoring incoming and outgoing packets to and from a software application;

placing ~~said~~ the software application on a zombie list or a watch list when a pattern of the incoming or outgoing packets to or from the software application~~s~~ matches that of the characteristics of a zombie application; and

blocking reception and transmission of packets to and from the software application when the software application has been placed on the watch list or the zombie list in a previous cycle and the software application further exhibits the characteristics of a zombie application.

23. (original) The computer program recited in claim 22, wherein the characteristics of a zombie application are transmitting a large number of packets while receiving no incoming packets.

24. (original) The computer program recited in claim 23, wherein the characteristics of a zombie application are receiving no incoming packets and having a zombie rating exceeding a predetermined value.

25. (currently amended) The computer program recited in claim 22, wherein ~~said~~ the zombie rating is based on the factors of whether the software application is an application or a process and whether the software application is user initiated or initiated at system startup.

26. (original) The computer program recited in claim 25, wherein the characteristics of a zombie application are rarely receiving incoming packets and having a zombie rating exceeding a predetermined value.

27. (currently amended) The computer program recited in claim 26, wherein ~~said~~ the zombie rating is based on the factors of whether the software application is an application or a process and whether the software application is user initiated or initiated at system startup.

28. (original) The computer program recited in claim 22, wherein the characteristics of a zombie application are receiving incoming packets at first and then not receiving or sending any packets and having a zombie rating exceeding a predetermined value.

29. (currently amended) The computer program recited in claim 28, wherein ~~said~~ the zombie rating is based on the factors of whether the software application is an application or a process and whether the software application is user initiated or initiated at system startup.

30. (new) The system of claim 1, wherein the incoming packets determined to be from the zombie application include request packets comprising target device information and a start sequence to enable the zombie application to begin executing, wherein the receive module discards the request packets before the request packets are allowed to

enter a network protocol module.

31. (new) The system of claim 1, wherein the receive module blocks the zombie applications from registering for network access via a network protocol module.

32. (new) A system for preventing denial of service attacks on a network, comprising:

a transmit module, coupled between a network protocol and a network interface, to receive outgoing packets from a software application via the network protocol and to discard the outgoing packets if the outgoing packets are determined to be from a zombie application, wherein the transmit module stops the transmission of the outgoing packets determined to be from the zombie application before the outgoing packets are allowed to enter the network interface for transmission over the network; and

a monitor module, in communication with the transmit module, to track packet transmission patterns from the software application and to determine whether the software application is the zombie application based upon the packet transmission patterns.

33. (new) The system of claim 32, the monitor module to determine that the software application is the zombie application by identifying that the software application is transmitting a large number of packets without receiving a large number of packets.

34. (new) The system of claim 33, wherein if the software application was

previously put on a zombie list and the monitor module determines that the software application is now receiving packets, the monitor module to remove the software application from the zombie list.

35. (new) The system of claim 32, the monitor module to determine that the software application is a possible zombie application by identifying that the software application is not receiving any packets and putting the software application on a watch list.

36. (new) The system of claim 35, the monitor module to determine a zombie rating for the software application if the software application is not a known good application, wherein the zombie rating is based on whether the software application runs as a process or an application and if the software program runs as an application, whether the software application was invoked by a user or launched at startup.

37. (new) The system of claim 32, further comprising:

a receive module, coupled between the network protocol and the network interface, to receive incoming packets from the network interface and to discard incoming packets related to a request for a zombie application, wherein the receive module blocks the request for the zombie application before the request enters the network protocol.

38. The system of claim 37, the monitor module in communication with the

receive module, wherein if the monitor module determines that the software application rarely receives incoming packets or that the software application previously received some packets and now is not sending packets or receiving more packets, the monitor module to place the software application on a watch list.

39. (new) The system of claim 38, the monitor module to determine a zombie rating for the software application if the software application is not a known good application, wherein the zombie rating is based on whether the software application runs as a process or an application and if the software program runs as an application, whether the software application was invoked by a user or launched at startup.

40. (new) An article comprising: a storage medium having a plurality of machine accessible instructions, wherein when the instructions are executed by a processor, the instructions provide for monitoring incoming and outgoing packets to and from a software application;

placing the software application on a zombie list or a watch list when a pattern of the incoming or outgoing packets matches characteristics of a zombie application; and

blocking reception of the incoming packets to the software application and blocking transmission of the outgoing packets to the network when the software application has been placed on the zombie list or the watch list and the software application continues to exhibit the characteristics of the zombie application.

41. (new) The article of claim 40, wherein the characteristics of a zombie

application are transmitting a large number of outgoing packets while receiving no incoming packets.

42. (new) The article of claim 40, wherein the characteristics of the zombie application include rarely receiving the incoming packets and having a zombie rating exceeding a predetermined value.

43. (new) The article of claim 42, wherein the zombie rating is based on whether the software application runs as a process or an application and if the software program runs as an application, whether the software application was invoked by a user or launched at startup.

44. (new) The article of claim 40, wherein the characteristics of the zombie application include having received some incoming packets then not sending any outgoing packets or receiving anymore incoming packets and having a zombie rating exceeding a predetermined value.

45. (new) The article of claim 44, wherein the zombie rating is based on whether the software application runs as a process or an application and if the software program runs as an application, whether the software application was invoked by a user or launched at startup.